基于BDD算法的故障诊断研究与应用

李淑英[1] 汪培桢[2] 杨 春[2]

（1. 南宁轨道交通集团有限责任公司 南宁 530029

2. 北京交通大学北京市轨道交通电气工程技术研究中心 北京 100044）



李淑英 女 1989年生，硕士研究生，主要研究方向为车辆通信与故障诊断技术。



汪培桢 女 1993年生，硕士，主要研究方向为电力电子与电力传动。

摘要：故障树分析法在运用过程中容易产生“维数爆炸”等问题，本文重点研究基于 BDD 算法的故障树分析法，分析故障树转化为 BDD 的方法，并基于 BDD 算法求解顶事件发生的概率以及底事件的结构重要度，最后采用该算法进行算例分析与应用。

关键词：故障树分析法 BDD 算法 故障诊断 结构重要度

中图分类号：TP274

Research and Application of Fault Diagnosis Based on BDD

Li Shuying[1] *Wang Peizhen*[2] *Yang Chun*[2]

（1. Nanning Rail Transit Co.,Ltd. Nanning 530021 China

2. Beijing Jiaotong University Beijing 100044 China）

Abstract: The process of applying fault tree analysis easily lead to "dimension explosion" and other issues. The BDD-based algorithm of fault tree analysis is studied in the paper, and the method of transformation from fault tree to BDD is analyzed. The top event probability and structural importance of the end of the event based on BDD algorithm is solved. Finally using this algorithm analyze and apply to example.

Keywords: Fault tree analysis, BDD algorithm, fault diagnosis, structure importance

联合基金项目“高速铁路电力牵引系统的安全性预测与控制（U1134204）”资助。

1　引言

故障树分析法（Fault Tree Anflysis，FTA）于1961年由美国贝尔实验室的科研人员首先提出，最先运用于火箭发射安全评估系统，取得了良好的效果。随后波音公司分别在定性和定量上发展了故障树分析法[1]。故障树分析法采用图形演绎的方式清晰表达系统的内在联系，直观性强、灵活性高。

故障树定性分析的目的是寻找故障树最小割集。最小割集是导致系统故障发生的基本原因的最小组合，描绘了系统的最薄弱环节。传统故障树定性分析最小割集主要采用下行法或上行法求解。但当系统故障树的复杂程度增加时，容易产生“组合爆炸”问题。为解决这一问题，可以采用BDD算法将故障树化成不交化形式，再做进一步分析[2]。

2　BDD算法

二元决策图（Binary Decision Diagram，BDD）[3]于1978年由Sheldon. B. Akers首先提出，被广泛应用于故障诊断和可靠性领域。BDD是将所有节点通过0或者1标识连接起来的有向无环图，通过简化布尔函数的Shannon分解树得到，其实质是最小割集的和[4]。

Shannon分解定理可以概括为：假设$f(x_1, x_2, \cdots, x_n)$是任一布尔函数，$x_i(i=1, 2, \cdots, n)$是$f(x)$的任一自变量，f_0表示x_i取值为0时的表达式，f_1表示x_i取值为1时的表达式，则$f(x)$可以表示为

$$f(x_1,x_2,\cdots,x_n)=x_i\cdot f_1+\overline{x}_i\cdot f_0 \tag{1}$$

其中，$\overline{x}_i$为x_i取非。

由式（1）可知，f_1和f_0都是布尔函数，运用Shannon分解定理将原函数分解到不能再进行为止，即可得到不交化结果。将故障树向BDD转化的过程需要使用ite（if-then-else）规则，其实质是Shannon分解树表达式的转化形式，ite（if-then-else）规则数学表达式为

$$ite(x_i,f_0,f_1)=x_i\cdot f_1+\overline{x}_i\cdot f_0 \tag{2}$$

采用ite（if-then-else）规则递归方式从故障树的最底事件开始逐层向上，每步置换均采用ite规则进行编码，直到使用底事件来替换所有逻辑门，最终求出顶事件的ite结构，即顶事件的BDD[5]。

假设$M=ite(x_i, F_1, F_0)$、$N=ite(x_j, G_1, G_0)$为两个BDD子结构，故障树底事件排序为$x_1 < x_2 < \cdots < x_n (x_i, x_j \in \{x_1, x_2, \cdots, x_n\})$，$M$和$N$经过规范后由含有“与”、“或”逻辑门和底事件构成，可以通过ite结构连接，连接过程遵守如下规则。

（1）　当$x_i=x_j$时，$M<OP>N=ite(x_i, F_1<OP>N_1, F_0<OP>N_0)$。

（2）　当$x_i<x_j$时，$M<OP>N=ite(x_i, F_1<OP>N, F_0<OP>N)$。

3　基于BDD算法的故障树顶事件发生概率及结构重要度分析

3.1　顶事件发生的概率

BDD分解后的结构函数积的和不相交，顶事件概率的求解过程不需要考虑化相交和为不交和的计算。求顶事件发生概率搜索是从根结点开始，通过从根结点向下回溯所有“1”终结点，写出故障树的对应不交化的表达式，再利用互斥事件概率和的算法求出顶事件发生的概率。

3.2　结构重要度分析

底事件结构重要度指部件在系统所处位置的重要程度，与部件本身故障概率无关[5]。通过故障树结构重要度分析，可对结构重要度较大的部件进行检测和试验，提前做出防范决策，对提高系统可靠性和安全性能有重要意义[6]。

若故障树中有n个底事件，第i个部件从正常状态（记为$x_i=0$）变为故障状态（记为$x_i=1$），其他基本事件状态不变，$\Phi(0_i, X)$表示底事件i处于正常状态（$x_i=0$），其他基本事件为任意情况时顶事件的状态逻辑值，$\Phi(1_i, X)$表示基本事件i处于故障状态（$x_i=1$），其他基本事件为任意情况时顶事件的状态逻辑值。

系统顶事件可能有以下四种情况：

（1）$\Phi(0_i, X)=0 \rightarrow \Phi(1_i, X)=1$

（2）$\Phi(0_i, X)=0 \rightarrow \Phi(1_i, X)=0$

（3）$\Phi(0_i, X)=1 \rightarrow \Phi(1_i, X)=1$

（4）$\Phi(0_i, X)=1 \rightarrow \Phi(1_i, X)=0$

以上四种情况中，情况（1）说明了底事件i从0到1的变化，导致顶事件状态从$\Phi(0_i, X)=0$变到$\Phi(1_i, X)=1$，其余的$n-1$个基本事件可能有2^{n-1}种状态。结构重要度将所有情况（1）的次数累加，再除以2^{n-1}，所得的数据用来表示基本事件i对系统故障的贡献大小程度。结构重要度计算公式为

$$I_i^{\Phi}=\frac{1}{2^{n-1}}n_i^{\Phi} \tag{3}$$

其中，$n_i^{\Phi} = \sum\left[\Phi(1_i, X) - \Phi(0_i, X)\right]$，$i = 1, 2, \cdots, n$。

4　基于 BDD 算法的故障诊断逻辑推理流程

故障树转化成 BDD 采用 ite（if-then-else）规则，为实现这一功能，先定义变量 ITE_str 和变量 BDD_str。变量 ITE_str 存储 ite 结构，包括 ite 结构名、结构中的三个元素，变量 BDD_str 存储二叉树形式的 BDD。故障树转化成 BDD 的运算流程如图 1 所示。

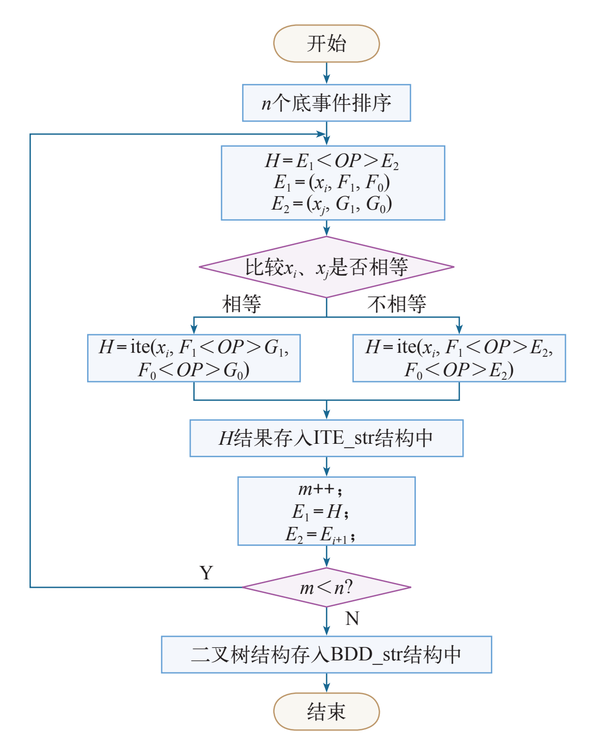


图 1　故障树转化成对应 BDD 的算法流程

Fig.1　Algorithm process of fault tree turn into its corresponding BDD

运用故障树分析法对系统进行故障诊断，可使用 BDD 算法先将故障树不交化，再进行定性分析和定量分析。诊断流程如图 2 所示。

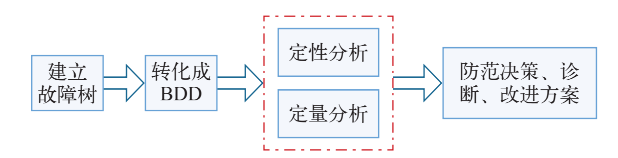


图 2　基于故障树分析法的故障诊断流程

Fig.2　Flowchart of fault diagnosis based on fault tree analysis method

5　基于 BDD 算法的牵引变流器故障诊断应用

牵引变流器综合逻辑判断实现列车牵引、制动功能，对其关键设备进行故障分析、提高系统可靠性、故障发生后快速定位故障原因并做出诊断措施尤为重要。如果出现模块故障，即进行快速诊断。牵引变流器模块正常工作温度在 20 ～ 85 ℃之间。图 3 所示为牵引变流器模块温度控制系统结构图，牵引变流器模块温度由温度传感器采集输入调理板，调理板将输入的电流信号转化成 DSP 可识别的电压信号，经过 A-D 采集模块进入 DSP 中换算成实际温度。若换算后的温度表明 IGBT 温度大于 35 ℃，则发出 D-A 转速控制信号，起动风机为牵引变流器模块散热，风机的起停状态通过 DI 反馈至 DSP 主控板。

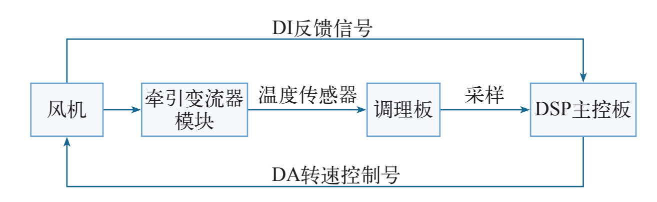


图 3　牵引变流器模块温度控制系统结构图

Fig.3　Temperature control system structure of traction converter module

牵引变流器模块过温故障为系统检测到模块温度高于 85 ℃，模块立即封脉冲，等待系统恢复故障才能重新启动。根据图 3 所示的温度控制模块系统结构图，建立系统故障树，列出牵引变流器模块过温故障的故障树事件见表 1。

牵引变流器模块温度控制系统功能进行分析后，建立故障树如图 4 所示。

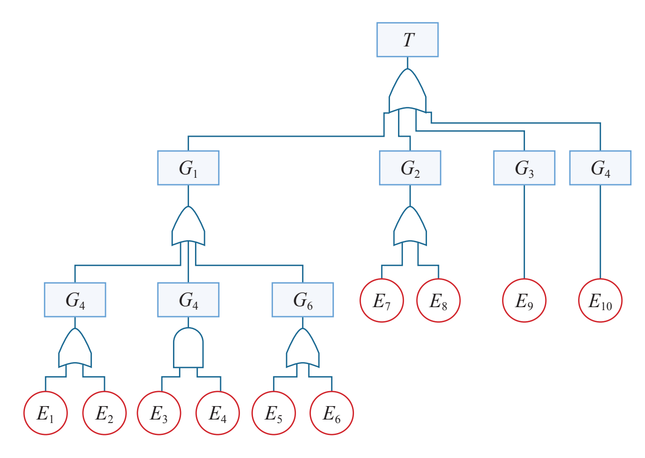


图 4　牵引变流器模块温度控制系统结构图

Fig.4　Temperature control system structure of traction converter module

从底层逐步使用 ite 结构换算，最终得到如图 5 所示故障树对应的 BDD。

从终结点开始，逐步向上搜索所有“1”分支，

表 1　故障树事件

Tab.1　Fault tree events

顶事件		中间事件				底事件	
编号	事件	编号	事件	编号	事件	编号	事件
T	DC-DC 模块过温故障	G_1	风机系统故障	G_4	风机信号故障	E_1	控制信号线路故障
						E_2	反馈信号线路故障
				G_5	风机故障	E_3	辅助供电系统故障导致谐波变大
						E_4	风机滤波装置故障
				G_6	风机线路故障	E_5	风机内部线路故障
						E_6	风机外部接线故障
		G_2	采样系统故障			E_7	调理板故障
						E_8	DSP 采样故障
		G_3	温度传感器故障			E_9	温度传感器故障
		G_4	辅助系统故障引起锁机			E_{10}	辅助故障引起系统锁机，风机无法起动

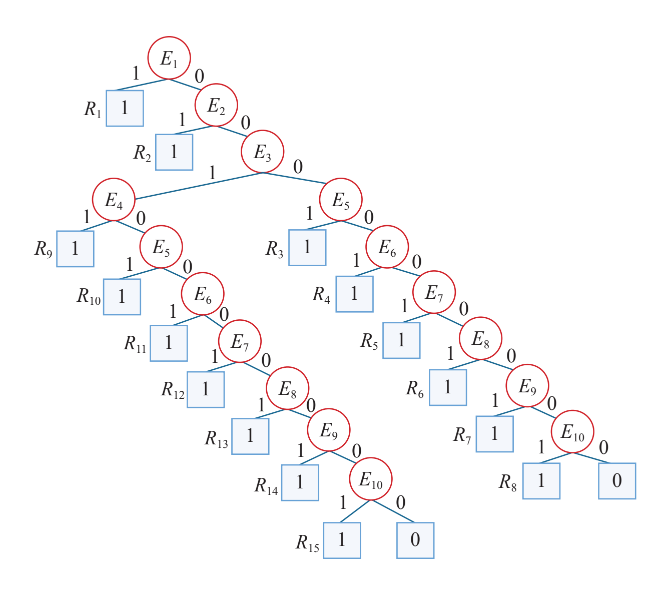


图 5　牵引变流器模块过温故障 BDD

Fig.5　BDD of traction converter module over temperature fault

代入结构式，牵引变流器模块过温故障全部最小割集为 $\{E_1\}$，$\{E_2\}$，$\{E_3, E_4\}$，$\{E_5\}$，$\{E_6\}$，$\{E_7\}$，$\{E_8\}$，$\{E_9\}$，$\{E_{10}\}$。

根据全部最小割集可知，除最小割集 $\{E_3, E_4\}$ 外，其他均为一阶最小割集，即其他底事件是系统的薄弱环节，每个底事件发生均会导致模块过温故障。在实际处理过程中，应该注意尽量降低一阶最小割集中事件发生的概率，例如，为调理板加上绝缘和抗干扰保护措施，降低调理板出现故障的概率。

根据 BDD 的性质可知，转化后的 BDD 反映了系统的所有故障模式，从根结点向下逐一搜索终结点为“1”的路径，就能写出顶事件发生的概率的表达式。将每一条路径用 F_i 来表达，则 F_i 的表达式见表 2。

表 2　F_i 表达式

Tab.2　Expression of F_i

$F_1 = E_1$	$F_2 = \bar{E}_1 E_2$
$F_3 = \bar{E}_1\bar{E}_2\bar{E}_3 E_5$	$F_4 = \bar{E}_1\bar{E}_2\bar{E}_3\bar{E}_5 E_6$
$F_5 = \bar{E}_1\bar{E}_2\bar{E}_3\bar{E}_5\bar{E}_6 E_7$	$F_6 = \bar{E}_1\bar{E}_2\bar{E}_3\bar{E}_5\bar{E}_6\bar{E}_7 E_8$
$F_7 = \bar{E}_1\bar{E}_2\bar{E}_3\bar{E}_5\bar{E}_6\bar{E}_7\bar{E}_8 E_9$	$F_8 = \bar{E}_1\bar{E}_2\bar{E}_3\bar{E}_5\bar{E}_6\bar{E}_7\bar{E}_8\bar{E}_9 E_{10}$
$F_9 = \bar{E}_1\bar{E}_2 E_3 E_4$	$F_{10} = \bar{E}_1\bar{E}_2 E_3\bar{E}_4 E_5$
$F_{11} = \bar{E}_1\bar{E}_2 E_3\bar{E}_4\bar{E}_5 E_6$	$F_{12} = \bar{E}_1\bar{E}_2 E_3\bar{E}_4\bar{E}_5\bar{E}_6 E_7$
$F_{13} = \bar{E}_1\bar{E}_2 E_3\bar{E}_4\bar{E}_5\bar{E}_6\bar{E}_7 E_8$	$F_{14} = \bar{E}_1\bar{E}_2 E_3\bar{E}_4\bar{E}_5\bar{E}_6\bar{E}_7\bar{E}_8 E_9$
$F_{15} = \bar{E}_1\bar{E}_2 E_3\bar{E}_4\bar{E}_5\bar{E}_6\bar{E}_7\bar{E}_8\bar{E}_9 E_{10}$	

牵引变流器模块过温故障的故障树结构函数不交化表达式为

$$T = \sum_{i=1}^{15} F_i \tag{4}$$

对故障树进行定量分析，需要根据大量的数据理论和实验得出系统各个设备的故障概率。本示例在缺少可靠数据情况下，采用假设数据来进行计算和验证。假设本示例中 $E_i (i = 1, 2, \cdots, 10)$ 这 10 个底事件的故障概率均为 0.2%，结合式（4），则有

$$p(T) = p(E_1) + p(\bar{E}_1)p(E_2) + \cdots + p(\bar{E}_1)p(\bar{E}_2)p(E_3)p(\bar{E}_4)p(\bar{E}_5)p(\bar{E}_6)p(\bar{E}_7)p(\bar{E}_8)p(\bar{E}_9)p(E_{10}) \approx 1.589\%$$

求底事件结构重要度，首先根据故障树结构容

易判断出 E_1、E_2、E_5、E_6、E_7、E_8、E_9、E_{10} 的结构重要度一样，E_3、E_4 的结构重要度相等。然后各个底事件统计满足 $\Phi(0_i, X) = 0 \rightarrow \Phi(1_i, X) = 1$ 判断条件的情况，根据式（3）即可计算各个底事件的结构重要度。

根据故障树最小割集及对应的 BDD 可知，E_3 取值从 0 变化到 1 引起系统状态从 0 变化到 1 的情况总共有一种，E_1 取值从 0 变化到 1 引起系统状态从 0 变化到 1 的情况总共有 3 种，E_3 和 E_1 引起的系统状态值改变分别见表 3、表 4。

表 3　E_3 引起系统状态值改变

Tab.3　System change the status value by E_3

底事件	E_3	E_1	E_2	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}
$\Phi(0_i, X)=0$	0	0	0	1	0	0	0	0	0	0
$\Phi 0_i, X)=1$	1	0	0	1	0	0	0	0	0	0

表 4　E_1 引起系统状态值改变

Tab.4　System change the status value by E_1

情况	底事件	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}
1	$\Phi(0_i, X)=0$	0	0	0	1	0	0	0	0	0	0
	$\Phi(0_i, X)=1$	1	0	0	1	0	0	0	0	0	0
2	$\Phi(0_i, X)=0$	0	0	0	0	0	0	0	0	0	0
	$\Phi(0_i, X)=1$	1	0	0	0	0	0	0	0	0	0
3	$\Phi(0_i, X)=0$	0	0	1	0	0	0	0	0	0	0
	$\Phi(0_i, X)=1$	1	0	1	0	0	0	0	0	0	0

根据以上分析可计算出底事件的结构重要度为

$$\begin{cases} I_1^{\Phi} = I_2^{\Phi} = I_5^{\Phi} = I_6^{\Phi} = I_7^{\Phi} = I_8^{\Phi} = I_9^{\Phi} = I_{10}^{\Phi} = \dfrac{3}{2^9} \\ I_3^{\Phi} = I_4^{\Phi} = \dfrac{1}{2^9} \end{cases}$$

由计算结果可知，E_1、E_2、E_5、E_6、E_7、E_8、E_9、E_{10} 的结构重要度大于 E_3 和 E_4，可见，前者这 8 个底事件对于系统发生牵引变流器模块过温故障的贡献程度更大，因此，在安排防范措施时优先考虑会有更好的效果。

通过对牵引变流器模块过温故障进行定性分析可知，系统的薄弱环节是一阶的割集，如果出现模块过温故障，优先考虑是薄弱环节引起，可以较快地定位故障原因，对系统的维修和升级有重要指导意义。通过对模块过温故障顶事件发生的概率计算可以预测系统出现故障的频率，本文使用的假设数据计算结果表明顶事件出现概率 1.589% 较高，在使用前应该多次对各个底事件进行故障排查，例如，可以通过线路选择、安装工艺等方面严格要求尽量避免出现外部线路故障。对于本例来说，结构重要度结果表明割集中一阶割集的底事件对系统故障贡献最高，还是以降低一阶割集底事件发生的概率对提高系统安全可靠性更重要。

6　结束语

本文重点研究基于 BDD 算法的故障诊断原理，包括故障树转化为 BDD、顶事件发生概率计算以及底事件结构重要度分析，并通过算例说明该方法的运用过程。结果表明，基于 BDD 算法的故障树分析法对故障诊断具有重要的指导意义。

参考文献

[1]　Marvin Rausand. System reliability theory: models, statistical methods, and applications[M]. 2nd ed. 郭强 , 等译 . 北京 : 国防工业出版社 , 2010.

[2]　徐忠昌. 保障性工程 [M]. 北京：兵器工业出版，2005.

[3]　Akers S. Binary decision diagrams[J]. IEEE Transactions on Computers, 1978, 27(6): 509-516.

[4]　陶勇剑，董德存，任鹏. 故障树分析的二元决策图方法 [J]. 铁路计算机应用，2009，18(9)：4-7. Tao Yongjian, Dong Decun, Ren Peng. Binary decision diagram of fault tree analysis[J]. Railway Computer Application, 2009, 18(9): 4-7.

[5]　Zhou H, et al. The study on fault tree analysis method for building height fall accident[J]. Coastal Engineering, 2008.

[6]　徐亨成，张建国. 基于 BDD 技术下的故障树重要度分析 [J]. 电子机械工程，2003 19(6)：1-4. Xu Hengcheng, Zhang Jianguo. The importance analysis of fault tree based on the binary decision diagram[J]. Mechatronic Engineering, 2003, 19(6): 1-4.